B.M.S. College of Engineering

Autonomous Institute, Affiliated to VTU

LAB REPORT

23CS3PCOOJ

Submitted in partial fulfillment of the requirements for Lab

Bachelor of Engineering

In

Computer Science and Engineering

Submitted by:

PREETHI NARASIMHAN

(1BM22CS207)

Department of Computer Science and Engineering,

B.M.S College of Engineering,

Bull Temple Road, Basavanagudi, Bangalore, 560 019

2023-2024

INDEX

Sl.No.	Title	Date
1	Complete scanned Observation Book	12/12/2023- 20/02/2024
2	Lab 1	12/12/2023
3	Lab 2	19/12/2023
4	Lab 3	26/12/2023
5	Lab 4	02/01/2024
6	Lab 5	09/01/2024
7	Lab 6	16/01/2024
8	Lab 7	23/01/2024
9	Lab 8	30/01/2024
10	Lab 9	06/02/2024
11	Lab 10	20/02/2024

WEEK-1 (Lab pgm 1)

Q6. Write a Java pgm to compute roots of a quadratic eqn $ax^2+bx+c=0$

```
import java.util.Scanner;
import java.lang.Math;
class Quadratic
{
    float a, b, c;
    double r1, r2, d;
    void getd()
    {
        Scanner s= new Scanner(System.in);
        System.out.println("Enter the
        coefficients of a, b, c");
        a = s.nextFloat();
        b = s.nextFloat();
        c = s.nextFloat();
    }
    void compute()
    {
        while(a==0)
        {
            System.out.println("Not a quadratic
            equation");
            System.out.println("Enter a
            non-zero value for a");
```

Date / /
Page

Next Lab : Create a class Book which contains 4 members : name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString() method that could display the complete details of the book. Develop a Java programme to create n book objects.

(Quadratic pgm continued)

```
Scanner s = new Scanner(System.in);
a = s.nextFloat();
}
d = b*b - 4*a*c;
if (d<0)
{ System.out.println("Roots are imagina
    r1 = -b/(2*a);
    r2 = Math.sqrt(-d)/(2*a);
    System.out.format("Root 1 = %.2
        + %.2Fi", r1, r2);
    System.out.format("Root 2 = %.2f
        - %.2fi", r1, r2);
}
else if(d>0)
{
r1 = (-b + Math.sqrt(d))/(2*a);
r2 = (-b - Math.sqrt(d))/(2*a);
System.out.println("Roots are real and
  distinct");
System.out.format("Root 1 = %.2f
```

Date __/__/__
Page _____

```
and Root 2 = %.2F", r1, r2);
        }
    else    // d=0 or Discriminant=0
        {
            r1 = (-b)/(2*a);
            System.out.println("Roots
            are real and equal");
            System.out.format("Root 1
            = Root 2 = %.2f", r1);
        }
    }
}
class QuadraticMain
{
    public static void main(String args[])
    {
        Quadratic q = new Quadratic();
        q.getd();
        q.compute();
    }
}
```

Output:

CASE (1) Enter the coefficients of a, b, c

1

-2

1

Roots are real and equal

Root 1 = Root 2 = 1.00

Date __/__/__
Page

CASE (2) Enter the coefficients of a, b, c
1
-5
6
Roots are real and distinct
Root 1 = 3.00 and Root 2 = 2.00

CASE (3) Enter the coefficients of a, b, c
7.2
5
9
Roots are imaginary
Root 1 = -0.35 + 1.06i
Root 2 = -0.35 - 1.06i

Date __/__/__
Page _____

WEEK 2 (Lab Pgm 2)

4. Write a pgm to compute SGPA of student.

```
import java.util.Scanner;
class Subject
{
        int subj_marks;
        int grade;
        int credits;
}
class Student
{
        String name;
        String usn;
        double SGPA;
        Scanner s;
        Subject subj[];  // array of objects.
        Student()
        {
            int i;
            subj = new Subject[8];
            for(i=0; i<8; i++)
                    subj[i] = new Subject();
            s = new Scanner(system.in);
        }
        void getStudentDetails()
        {
                System.out.print("Enter Student
                  name: ");
                name = s.next();
                System.out.print("Enter USN: ");
                usn = s.next();
        }
```

```
void getMarks()
{   S = new Scanner(System.in)
    for(int i=0; i<8; i++)
    {
        System.out.print("Enter details for ~~studen~~ subject "+(i+1)+":");
        System.out.print("Enter marks of student : ");
        subj[i].subj_marks = S.nextInt();
        System.out.print("Enter credits of subject : ");
        subj[i].credits = S.nextInt();
        if(subj[i].subj_marks>=90)
        {
            subj[i].grade = 10;
        }
        else if(subj[i].subj_marks)=80)
        {
            subj[i].grade = 9;
        }
        else if(subj[i].subj_marks>=70)
        {
            Subject[i].grade = 8;
        }
        else if(subj[i].subj_marks>=60)
        {
            subj[i].grade = 7;
        }
        else if(subj[i].subj_marks>=50)
        {
            subj[i].grade = 6;
        }
        else if(subj[i].subj_marks>=40)
```

```
            {
                subj[i].grade = 5;
            }
            else
            {
                subj[i].grade = 0;
            }
        }
    }
    void computeSGPA()
    {
        double totalcredits = 0;
        double weightedSum = 0;
        for(int i = 0; i<8; i++)
        {
            totalcredits += subj[i].credits;
            weightedSum += subj[i].grade *
                           subj[i].credits;
        }
        SGPA = weightedsum / totalcredits;
    }
    void DisplayDetails()
    {
        System.out.println("\n Student Details:");
        System.out.println("Name: " + name);
        System.out.println("USN: " + usn);
        System.out.println("SGPA: " + SGPA);
    }
}
public class StudentMain {
    public static void main(String[] args){
        Student s1 = new Student();
        s1.getStudentDetails();
```

Date
Page

```
        s1.getMarks();
        s1.computeSGPA();
        s1.DisplayDetails();
    }
```

O/p:

```
Enter Student Name: Preethi
Enter USN: 1BM22CS207
Enter details for subject 1:
Enter Marks: 75
Enter Credits: 4
Enter details for subject 2:
Enter Marks: 78
Enter Credits: 4
Enter details for subject 3:
Enter Marks: 94
Enter credits: 3
Enter details for subject 4:
Enter Marks: 67
Enter credits: 3
Enter details for subject 5:
Enter Marks: 87
Enter credits: 3
Enter details for Subject 6:
Enter Marks: 100
Enter credits: 1
Enter details for Subject 7:
Enter credits: 1
Enter Marks: 100
Enter details for Subject 8:
Enter Marks: 100
Enter Credits: 1
```

~~Quadratic~~

Student Details:

Name : Preethi

USN: 1BM22CS207

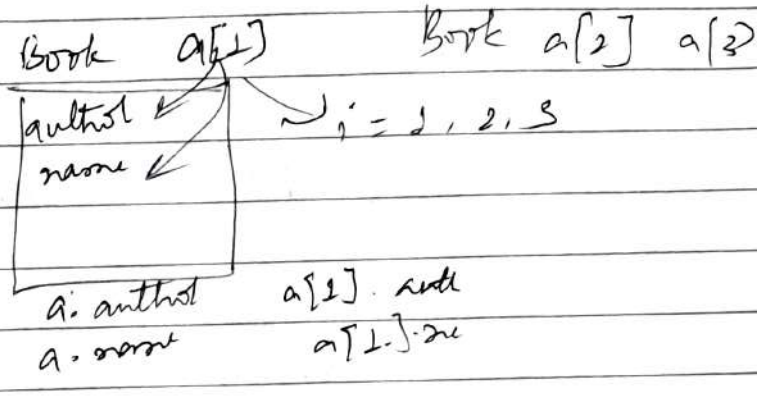
SGPA : 8.6

26/12/23

Date __/__/__
Page _____

WEEK-3 (Lab pgm 3)

3. Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set & get the details of the objects. Include a toString() method that could display the complete details of the book. Develop a Java pgm to create n book objects



Pgm:

```
import java.util.Scanner;
class Books
{
    String name;
    String author;
    int price;
    int num_pages;
    Books (String name, String author,
           int price, int num_pages)
    {
        this.name = name;
        this.author = author;
        this.price = price;
        this.num_pages = num_pages;
    }
```

```
public String toString()
{
    String name, author, price, num_pages;
    name = "Book name: " + this.name + "\n";
    price = "Price: " + this.price + "\n";
    num_pages = "Number of pages: " +
                this.num_pages + "\n";
    return name + author + price + num_pages;
}
}
class Main
{
    public static void main(String args[])
    {
        Scanner s = new Scanner(System.in);
        int n;
        String name;
        String author;
        int price;
        int num_pages;
        System.out.println("Enter the no. of
        books: ");
        n = s.nextInt();
        Books [] b = new Books[n];
        System.out.println("Enter details of"
                + n + "books: ");
        for (int x = 0; x < n; x++)
        {
            System.out.println("Enter the
            name of book " + (x+1) + ": ");
            name = s.next();
            System.out.println("Enter author
            of book " + (x+1) + ": ");
```

```
        author = s.next();
        System.out.println(" Enter
        the perice of book "+(i+1)+":");
        price = s.nextInt();
        System.out.println(" Enter
        no. of pages of book "+(i+1)+
        ":");
        num_pages = s.nextInt();
        b[i] = new Books(name,
        author, price, num_pages);
    }
    System.out.println("Details of "+n+"
            books:");
    for(int i=0; i<n; i++)
    {
        System.out.println(b[i].toString
            ());
    }
  }
}
```

OUTPUT:

```
Enter the no. of books
3
Enter details of 3 books:
Enter the name of book 1:
C
Enter the author of book 1:
Balaguruswamy
Enter the price of book 1:
100
Enter the no. of pages of book 1:
200
```

Enter the name of ~~bok~~ book 2:
C++
Enter the author of book 2:
PReddy
Enter the price of book 2:
200
Enter the no. of pages of book 2:
200
Enter the name of book 3:
Java
Enter the author of book 3:
Swamy
Enter the price of book 3:
300
Enter the no. of pages of book 3:
350
Details of 3 books:
Book name: C
Author name: Balaguruswamy
Price: 100
Number of pages: 200

Book name: C++
Author name: PReddy
Price: 200
Number of pages: 200

Book name: Java
Author name: Swamy
Price: 300
Number of pages: 350

24/12/21

Java - OOJ Lab

Lab - 4

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea() that prints the area of the given shape.

```
import java.util.Scanner;
abstract class InputScanner
{
        abstract void get_dim();
}
abstract class Shape extends InputScanner
{
    int a, b;
    Shape(int a, int b)
    {
        this.a = a;
        this.b = b;
    }
    Shape(int a)
    {
        this.a = a;
        this.b = 0;
    };
    abstract void printArea();
}
class Rectangle extends Shape
{
    Rectangle(int a, int b)
```

```
    {
        super(a, b);
    }
    void get_dim()
    {
        System.out.println("Enter the
        dimensions of the rectangle
        (length and breadth)");
        Scanner s= new Scanner
                    (System.in);
        a=s.nextInt();
        b= s.next Int();
    }
    void printArea()
    {
        System.out.println("Area of
        Rectangle = "+(a*b));
    }
}
class Triangle extends Shape
{
    Triangle(int a, int b)
    {
        super(a, b);
    }
    void get_dim()
    {
        System.out.println("Enter the
        dimensions of the triangle (base and
        height)");
        Scanner s= new Scanner(System.in);
        a= s.nextInt();
        b= s.next Int();
    }
void printArea()
```

```
        {
            System.out.println("Area of
            triangle = " + (0.5*a*b));
        }
}
class Circle extends Shape
{
    Circle(int a)
    {
        super(a);
    }
    void get_dim()
    {
        System.out.println("Enter the
        dimensions of the circle(radius)");
        Scanner s = new Scanner(System.in);
        a = s.nextInt();
    }
    void printArea(){
    {
        System.out.println("Area of circle = " +
            (3.14*a*a));
    }
}
class ShapeMain
{
    public static void main(String args[])
    {
        Rectangle r = new Rectangle(0,0);
        Triangle t = new Triangle(0,0);
        Circle c = new Circle(0);
        r.get_dim();
        t.get_dim();
        c.get_dim();
        r.printArea();
```

```
        t.printArea();
        c.printArea();
    }
}
```

o/p:

Enter dim of Rectangle (length and breadth):
2 3
Enter dim of Triangle (base and height):
2 4
Enter dim of Circle (radius):
3
Area of rectangle = 6
Area of Triangle = 4.0
Area of Circle = 28.2599999

02.01.24

WEEK-5

9/1/24

Date __/__/__
Page _____

[MODIFIED PROGRAM, REFER THIS]

LAB- Program 5

Bank (Savings & current Account)

```
import java.util.Scanner;
class Account
{
    String cust_name;
    int Accno;
    double balance = 0;
    public void get_details()
    {
        Scanner scanner = new Scanner(System.in);
        System.out.println(" Enter the customer
        name: ");
        cust_name = scanner.next();
        System.out.println(" Enter the account
        number: ");
        Accno = scanner.nextInt();
    }

    public void display_details(String Acc_type)
    {
        System.out.println(" Customer Name: "+
            cust_name);
        System.out.println(" Account Number: "+
        Accno);
        System.out.println(" Account type: "+
        Acc_type);
        System.out.println(" Balance : " + balance);
    }
}
```

Date __/__/__
Page _____

```
class Sav_acc extends Account
{
    double interestRate = 0.05;

    public Sav_acc(double balance)
    {
        this.balance = balance;
    }

    public void deposit_s()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the deposit
        amount : ");
        double amount = s.nextDouble();
        balance += amount;
        System.out.println("Deposit of "+ amount
        + " successful.");
    }

    public void computeInterest()
    {
        double interest = balance * interestRate;
        balance += Interest;
        System.out.println("Interest of "+
        interest + " computed and added to
        the account.");
    }

    public void withdraw_s()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the withdrawal
```

```
            amount: ");
            double amt = s.nextDouble();
            if (balance >= amt)
            {
                balance -= amt;
                System.out.println("Withdrawal
                of " + amt + " successful");
            }
            else
            {
                System.out.println("Insufficient
                funds. Withdrawal not
                permitted.");
            }
        }
}
class Cur_acc extends Account
{
    public Cur_acc(double balance)
    {
        this.balance = balance;
    }

    public void deposit_c()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the deposit
        amount: ");
        double amount = s.nextDouble();
        balance += amount;
        System.out.println("Deposit of " +
        amount " + amount + " successful.");
}
```

```
public void Cheque_book()
{
    String phone_no;
    String address;
    Scanner s = new Scanner(System.in);
    System.out.println("Enter your
    phone number:");
    phone_no = s.next();
    System.out.println("Enter your
    address :");
    address = s.next();
    System.out.println("cheque book
    request processed successfully.");
}

public void check_min_bal()
{
    double min_bal = 1000;
    double service_charge = 50;
    if (balance < min_bal)
    {
        balance -= service_charge;
        System.out.println("Service charge
        of " +service_charge + " imposed
        due to balance falling below
        minimum.");
    }
    else
    {
      System.out.println("Current balance is valid");
    }
}
```

```
public void withdraw_c()
{
    Scanner s= new Scanner(System.in);
    System.out.println(" Enter the
    withdrawal amount ");
    double amt = s.nextDouble();
    if (balance >= amt)
    {
      Syst
      balance -= amt;
      System.out.println("Withdrawal of
        " + amt + " Successful");
    }
    else
    {
      System.out.println(" Insufficient
      funds. Withdrawal not permitted.");
    }
  }
}
class BankAK // Execution starts here
{
  public static void main(String [] args)
  {
    Scanner p = new Scanner(System.in);
    System.out.println("Enter the type
    of account (Savings or Current): ");
    String Acc_type = p.next();

valid");    if (Acc_type.equals("Savings")
    {
      Sav_acc sa = new Sav_acc(0);
      sa.get_details();
```

```
int choice;
do {
System.out.println("\t\t MENU\t\t");
System.out.println(" 1. Deposit\t
2. Withdraw\t 3. Compute interest \t
4. Display Account details \t
5. Exit ");
System.out.println(" Enter choice: ");
choice = p.nextInt();
switch (choice)
{
    case 1:
        sa.deposit_s();
        break;
    case 2:
        sa.withdraw_s();
        break;
    case 3:
        sa.computeInterest();
        break;
    case 4:
        sa.display_details(Ace_type);
        break;
    case 5:
        System.out.println("
          Exiting...");
        break;
    default:
        System.out.println(" Invalid
        choice ");
}
} while (choice != 5);
}
```

Date __/__/__
Page _____

```
else if(Acc_type.equals("Current"))
{
    Cur_ac ca = new Cur_acc(0);
    ca.get_details();
    int choice;
    do{
    System.out.println("\t\t MENU \t\t");
    System.out.println(" 1. Deposit \t
    2. Withdrawal \t 3. Avail Cheque book \t
    4. Check min balance \t 5. Display
    details \t 6. Exit ");
    System.out.println("Enter choice: ");
    choice = p.nextInt();
    switch(choice)
    {
        case 1: ca.deposit_c();
                break;
        case 2: ca.withdraw_c();
                break;
        case 3: ca.Cheque_book();
                break;
        case 4:
                ca.check_min_bal();
                break;
        case 5:
                ca.display_details(Acc_type);
                break;
        case 6:
                System.out.println("Exiting...");
                break;
        default:
                System.out.println(" Invalid
                choice");
    }
```

```
        }while(choice!=6);
    }
    else
    {
        System.out.println("Invalid account type");
    }
}
}
```

OUTPUT:

```
Enter the type of account(Savings or Current):
Savings
Enter the customer name:
Krishna
Enter the account number:
12345

                MENU
1. Deposit  2. Withdraw  3. Compute interest
4. Display Account details  5. Exit
Enter choice:
1
Enter the deposit amount:
2000
Deposit of 2000.0 successful

                MENU
1. Deposit  2. Withdraw  3. Compute interest
4. Display Account details  5. Exit
Enter choice:
2
Enter the withdrawal amount:
200
Withdrawal of 200.0 successful
```

Date __/__/__
Page _____

MENU

1. Deposit 2. Withdraw 3. Compute interest
4. Display Account details 5. Exit
Enter choice:
3
Interest of 90.0 computed and added to ~~amount~~ account.
Enter choice:
4
Customer Name: Krishna
Account Number: 12345
Account Type: Savings
Balance: 1890.0

MENU

1. Deposit 2. Withdraw 3. Compute interest
4. Display Account details 5. Exit
Enter choice:
5
Exiting...

09/01/24

Date __/__/__
Page _____

WEEK-6 Pgm. 23/1/24.

Student.java

```
package CIE;
import java.util.Scanner;
class Student
{
    protected String usn = new String();
    protected String name = new String();
    protected int sem;
    public void inputStudentDetails()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the usn");
        usn = s.next();
        System.out.println("Enter the name");
        name = s.next();
        System.out.println("Enter the semester");
        sem = s.nextInt();
    }
    public void displayStudentDetails()
    {
        System.out.println("USN : " + usn);
        System.out.println("Name is : "+name);
        System.out.println("Semester " + sem);
    }
}
```

Internals.java

```
package CIE;
import java.util.Scanner;
public class Internals extends Student
{
    protected int marks[] = new int[5];
    public void inputCIEmarks()
    {
        Scanner s = new Scanner(System.in);
        System.out.println("enter the
        marks of 5 subjects..");
        for(int i=0; i<5; i++)
        {
            System.out.println("enter the
            marks of 5 subjects:");
            for(int i=0; i<5; i++)
            {
                System.out.println("
                Subject "+(i+1)+": ");
                marks[i] = s.nextInt();
            }
        }
    }
}
```

Externals.java

```
package SEE;
import CIE.Internals;
import java.util.Scanner;
public class Externals extends Internals
{
    protected int marks[];
    protected int finalMarks[];
    public Externals()
    {
        marks = new int[5];
        finalMarks = new int[5];
    }
    public void inputSEEmarks()
    {
        Scanner s = new Scanner(System.in);
        for(int i=0; i<5; i++)
        {
            System.out.println("Subject"
            +(i+1)+"marks:");
            marks[i] = s.nextInt();
        }
    }
    public void calculateFinalMarks()
    {
        for(int i=0; i<5; i++)
        {
            finalMarks[i] = marks[i]/2
                + Super.marks[i];
        }
    }
    public void displayFinalMarks()
    {
        displayStudentDetails();
```

```
        for(int i = 0; i < 5; i++)
        {
            System.out.println("Subject " +
                (i+1) + " : " + finalMarks[i]);
        }
    }
}
```

Main.java

```
import SEE.Externals;
class Main {
    public static void main(String args[])
    {
        int numOfStudents = 2;
        Externals finalMarks[] = new Externals
                                    [numOfStudents];
        for(int i = 0; i < numOfStudents; i++)
        {
            finalMarks[i].calculateFinalMarks();
            finalMarks[i].displayFinalMarks();
        }
    }
}
```

Date __/__/__
Page _____

O/p:

Enter the usn
2023BMS02608
Enter the name
Preethi
Enter the semester
3
Enter CIE marks
Enter the marks for 5 subjects
Subject 1 : ~~89~~ 24
Subject 2 : ~~98~~ 27
Subject 3 : 16
Subject 4 : 28
Subject 5 : 29

Enter SEE marks
Subject 1 marks : 89
Subject 2 marks : 98
Subject 3 marks : 88
Subject 4 marks : 87
Subject 5 marks : 86

USN : 2023BMS02608
NAME IS : Preethi
Semester : 3
Subject 1 : 68
Subject 2 : 76
Subject 3 : 60
Subject 4 : 71
Subject 5 : 72

WEEK-7

30/1/24

WEEK-7

```
import java.util.Scanner;
class WrongAge extends Exception
{
    public WrongAge()
    {
        super("Age Error");
    }
    public WrongAge(String message)
    {
        super(message);
    }
}
class InputScanner
{
    protected Scanner s;
    public InputScanner()
    {
        s = new Scanner(System.in);
    }
}
class Father extends InputScanner
{
    protected int fatherAge;
    public Father() throws WrongAge
    {
        System.out.println("Enter the Father's age:");
        fatherAge = s.nextInt();

        if(fatherAge < 0)
        {
            throw new WrongAge("Age
        }       cannot be negative");
```

```
    }
    public void display()
    {
        System.out.println("Father's
        Age : " + FatherAge);
    }
}
class Son extends Father
{
    private int sonAge;
    public Son() throws WrongAge
    {
        super();
        System.out.println("Enter
        Son's age : ");
        sonAge = s.nextInt();
        if (sonAge >= fatherAge)
        {
            throw new WrongAge("Son's
        age cannot be greater than fatherus's
        age");
        else if (sonAge < 0)
        {
            throw new WrongAge("
        Age cannot be negative");
        }
    }
    public void display()
    {
        super.display();
        System.out.println("Son's Age : " +
        sonAge);
    }
}
```

```
public class Age
{
    public static void main(String[] args)
    {
        try
        {
            Son son = new Son();
            son.display();
        }
        catch(WrongAge e)
        {
            System.out.println("Exception:"
            + e.getMessage());
        }
    }
}
```

OUTPUT:

(1) Enter the father's age:
35
Enter Son's age:
-12
Exception: Age cannot be negative

(2) Enter the father's age:
40
Enter Son's age:
45
Exception: Son's age cannot be greater than father's age.

30/01/24

Date __/__/__
Page _____

WEEK-8 6/2/24

Lab pgm 8:

Write a Java pgm which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

```
import java.util.Scanner;
class DisplayThread extends Thread
{
    private String message;
    private int interval;
    public DisplayThread (String message,
    int interval)
    {
        this.message = message;
        this.interval = interval;
    }
    public void run()
    {
        while(true)
        {
            System.out.println(message);
            try {
                Thread.sleep(interval * 1000);
            }
            catch (InterruptedException e)
            {
                e.printStackTrace();
            }
        }
    }
}
```

```
public class ThreadExample
{
    public static void main (String [] args)
    {
        DisplayThread thread1 = new
                DisplayThread ("BMS
                College of Engineering", 10);
        DisplayThread thread2 = new DisplayThread
        ("CSE", 2);
        thread1.start();
        thread2.start();
    }
}
```

O/P:

BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
BMS College of Engineering

6/2/20

WEEK - ~~10~~ 9

(Lab Pgm 10)

Inter Process Communication

Incorrect Implementation of a producer and consumer

```
class Q
{
    int n;
    synchronized int get()
    {
        System.out.println("Got : "+n);
        return n;
    }
    synchronized void put(int n)
    {
        this.n = n;
        System.out.println("Put : "+n);
    }
}
class Producer implements Runnable
{
    Q q;
    Producer(Q q)
    {
        this.q = q;
        new Thread(this, "Producer").start();
    }
    public void run()
    {
        int i=0;
```

```
        while (i < 10)
        {
            q.put(i++);
        }
    }
}
class Consumer implements Runnable
{
    Q q;
    Consumer(Q q)
    {
        this.q = q;
        new Thread(this, "Consumer").start();
    }
    public void run()
    {
        int i = 0;
        while (i < 10)
        {
            int r = q.get();
            i++;
        }
    }
}
class PC
{
    public static void main(String args[])
    {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C
        to stop.");
    }
```

```
}
```

O/P:

```
Press Control-C to stop.
Put: 0
Put: 1
Put: 2
Put: 3
Put: 4
Put: 5
Put: 6
Put: 7
Put: 8
Put: 9
Got: 9
Got: 9
Got: 9
Got: 9
Got: 9
Got: 9
Got: 9
Got: 9
Got: 9
Got: 9
```

Date __/__/__
Page _____

Correct implementation of a producer and consumer

```
class Q
{
    int n;
    boolean valueSet = false;
    synchronized int get()
    {
        while(!valueSet)
        try
        {
            System.out.println("\n
            Consumer waiting \n");
            wait();
        }
        catch(InterruptedException e)
        {
            System.out.println("
            InterruptedException caught");
        }
        System.out.println("Got : "+n);
        valueSet = false;
        System.out.println("\n Intimate
        Producer\n");
        notify();
        return n;
    }
    synchronized void put(int n)
    {
        while(valueSet)
```

```
.. try
   {
      System.out.println("\n
      Producer waiting \n");
      wait();
   }
   catch (InterruptedException e)
   {
      System.out.println("Interrupted
      Exception caught");
   }
   this.n = n;
   valueSet = true;
   System.out.println("Put: "+n);
   System.out.println("\n Intimate
   Consumer \n");
   notify();
  }
}
class Producer implements Runnable
{
   Q q;
   Producer(Q q)
   {
      this.q = q;
      new Thread(this, "Producer").start();
   }
   public void run()
   {
      int i = 0;
      while(i < 10)
      {
         q.put(i++);
```

```
        }
    }
}
class Consumer implements Runnable
{
    Q q;
    Consumer(Q q)
    {
        this.q = q;
        new Thread(this, "Consumer").start();
    }
    public void run()
    {
        int i=0;
        while(i<10)
        {
            int r = q.get();
            System.out.println("consumed:
            "+r);
            i++;
        }
    }
}
class PCFixed
{
    public static void main(String args[])
    {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press
        Control-C to stop.");
    }
}
```

O/P:

press control-C to stop
Put : 0
Intimate Consumer
Producer waiting
Got : 0
Intimate Producer
Put : 1
Intimate Consumer
Producer waiting
consumed : 0
Got : 1
Intimate Producer
consumed : 1
Put : 2
Intimate Consumer
Producer waiting
Got : 2
Intimate Producer
consumed : 2
Put : 3
Intimate Consumer
Producer waiting
Got : 3
Intimate Producer
consumed : 3
Put : 4
Intimate Consumer
Got : 4
Intimate Producer
consumed : 4

Date __/__/__
Page _____

Deadlock

```
class A
{
    synchronized void foo(B b)
    {
        String name = Thread.currentThread
                      ().getName();
        System.out.println(name + " entered
          A.foo");
        try
        {
            Thread.sleep(1000);
        }
        catch(Exception e)
        {
            System.out.println("A Interrupted");
        }
        System.out.println(name + " trying
          to call B.last()");
        b.last();
    }
    void last()
    {
        System.out.println("Inside A.last");
    }
}
class B
{
    synchronized void bar(A a)
    {
        String name = Thread.currentThread
                      ().getName();
```

```
System.out.println(name + "entered B.last");
try
{
    Thread.sleep(1000);
}
catch(Exception e)
{
    System.out.println("B Interrupted
    ~~Inside A.last~~");
}
System.out.println(name + "trying to call A.last()");
a.last();
}
void last()
{
    System.out.println("Inside A.last");
}
}
class Deadlock implements Runnable
{
    A a = new A();
    B b = new B();
    Deadlock()
    {
        Thread.currentThread().setName("MainThread");
        Thread t = new Thread(this,
        "RacingThread");
        t.start();
        a.foo(b);
        System.out.println("Back in main thread");
    }
```

```
public void run()
{
    b.bar(a);
    System.out.println("Back
    in other thread");
}
public static void main(String
 args[])
{
    new Deadlock();
}
}
```

O/P:

```
Main Thread entered A.foo
Racing Thread entered B.bar
Main Thread trying to call B.last()
Inside A.last
Back in main thread
Racing Thread trying to call A.last()
Inside A.last
Back in other thread
```

read");

thread");

13/2/24

Date
Page

WEEK - 10
(Lab pgm 9)

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
class SwingDemo
{
    SwingDemo()
    {
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        jfrm.setDefaultCloseOperation
            (JFrame.EXIT_ON_CLOSE);
        JLabel jlab = new ~~Jlab~~ JLabel
                ("Enter the ~~divider~~ dividend and
                ~~dividend~~ divider");
        JTextField ajtf = new JTextField(8);
        JTextField ajtf = new JTextField(8);
        JButton button = new JButton("Calcu");

        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();

        jfrm.add(err);
        jfrm.add(jlab);
        jfrm.add(ajtf);
        jfrm.add(bjtf);
        jfrm.add(button);
```

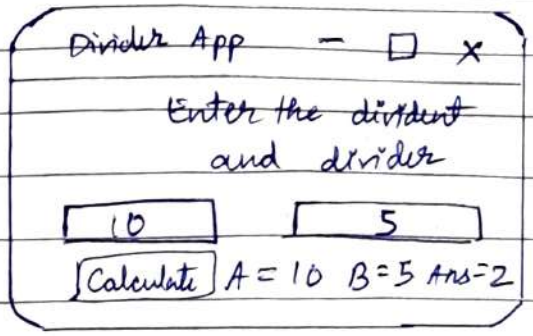
```
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);

ActionListener l = new
ActionListener()
{
    public void actionPerformed
    (ActionEvent evt)
    {
        System.out.println("Action
        event from a text field");
    }
};
ajtf.addActionListener(l);
bjtf.addActionListener(l);
button.addActionListener(new
ActionListener()
{
    public void actionPerformed
    (ActionEvent evt)
    {
        try {
            int a = Integer.
            parseInt(ajtf.getText
            ());
            int b = Integer.
            parseInt(bjtf.
            getText());
            int ans = a/b;

            alab.setText("\n A = "+a);
            blab.setText("\n B = "+b);
```

```
            anslab.setText("\n Ans = "+ans);
        }
        catch(NumberFormatException e)
        {
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("Enter Only Integers!");
        }
        catch(ArithmeticException e)
        {
            alab.setText("");
            blab.setText("");
            anslab.setText("");
            err.setText("B should be NON zero!");
        }
    }
});
jfrm.setVisible(true);
}
public static void main(String args[])
{
    SwingUtilities.invokeLater(new Runnable()
    {
        public void run()
        {
            new SwingDemo();
        }
    });
}
}//end of class Swing Demo
```

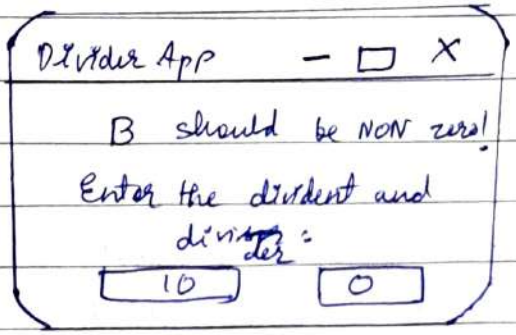
Date __/__/__
Page _____

OUTPUT:

i)



ii)



Definition of the following fns used in the program:

⇒ 1) JFrame: It is a class that represents the window containing the GUI.

⇒ 2) setSize: Is a method of 'JFrame' class that sets the size of the frame. (275 × 158 width height)

⇒ 3) setLayout: Is a method of the 'Container' used to set layout manager for the container.

⇒ 4) setDefaultCloseOperation: Is a method of 'JFrame' which sets default operation when the frame is closed. Here 'JFrame.EXIT_ON_CLOSE' is set, ie the application will terminate when frame is closed.

⇒ 5) JLabel: It is class used to display a non-edible text or image. Here it is used to display "Enter dirend and divider"

Date / /
Page

Select Table1 as A, Tab 2 as B, Tab3 as C
Tab4 as D
A.Tab1 = B.Tab2 and B.Tab2 = C.Tab3
and C.Tab3 = D.Tab4
where na = ' ' and ge ( )
w1 (a) = 2;

Stabalie
x Join
8 won con'ct

⇒ 6) JTextField : It is a class used to create a text field component that allows the user to enter text.

⇒ 7) add: Is a method of 'Container' class used to add components to the container. Here, 'JLabel', 'JTextField', 'JButton' are added to the frame using 'add' methods.

⇒ 8) ActionListener: Is an interface used to handle action events. Here, action listeners are added to 'aftf', 'bjtf' and 'button' to perform certain actions when specific events occur.

⇒ 9) setText: Is a method of 'JLabel' class used to set the text of the label dynamically. In this program, 'setText' is used to update the labels 'alab', 'blab', 'anslab' with calculated values or error messages.

20.02.24

Lab 1

Develop a Java program that prints all real solutions to the quadratic equation ax 2 +bx+c = 0. Read in a, b, c and use the quadratic formula. If the discriminate b^2 – 4ac is negative, display a message stating that there are no real solutions.

```
import java.util.Scanner;
import java.lang.Math;
class Quadratic
{
        float a,b,c;
        double r1,r2,d;
        void getd()
        {
          Scanner s=new Scanner(System.in);
          System.out.println("Enter the coefficients of a,b,c");
          a=s.nextFloat();
          b=s.nextFloat();
          c=s.nextFloat();
        }
        void compute()
        {
           while(a==0)
            {
```

```
        System.out.println("Not a quadratic equation");
        System.out.println("Enter a non-zero value for a");
                Scanner s=new Scanner(System.in);
                a=s.nextFloat();
}
d=b*b-4*a*c;
if(d==0)
{
        r1=(-b)/(2*a);
                System.out.println("Roots are real and equal");
        System.out.format("Root 1= Root 2= %.2f",r1);
}
else if(d<0)
{
        System.out.println("Roots are imaginary");
        r1=-b/(2*a);
        r2=Math.sqrt(-d)/(2*a);
        System.out.format("Root 1= %.2f+%.2fi",r1,r2);
        System.out.format("Root 2= %.2f-%.2fi",r1,r2);
}
else
{
                r1=(-b+Math.sqrt(d))/(2*a);
```

```
                    r2=(-b-Math.sqrt(d))/(2*a);
                System.out.println("Roots are real and distinct");
                System.out.format("Root 1= %.2f and Root 2= %.2f",r1,r2);
            }
        }
}
class QuadraticMain
{
        public static void main(String args[])
        {
            Quadratic q=new Quadratic();
            q.getd();
            q.compute();
        }
}
```

Output:

```
C:\Users\preet\OneDrive\Desktop\3rd Sem Java Pgms>java QuadraticMain
Enter the coefficients of a,b,c
1 2 1
Roots are real and equal
Root 1= Root 2= -1.00
C:\Users\preet\OneDrive\Desktop\3rd Sem Java Pgms>javac QuadraticMain.java

C:\Users\preet\OneDrive\Desktop\3rd Sem Java Pgms>java QuadraticMain
Enter the coefficients of a,b,c
5 1 1
Roots are imaginary
Root 1= -0.10+0.44iRoot 2= -0.10-0.44i
C:\Users\preet\OneDrive\Desktop\3rd Sem Java Pgms>javac QuadraticMain.java

C:\Users\preet\OneDrive\Desktop\3rd Sem Java Pgms>java QuadraticMain
Enter the coefficients of a,b,c
1 10 2
Roots are real and distinct
Root 1= -0.20 and Root 2= -9.80
C:\Users\preet\OneDrive\Desktop\3rd Sem Java Pgms>
```

Lab 2

Develop a Java program to create a class Student with members usn, name, an array credits and an array marks. Include methods to accept and display details and a method to calculate SGPA of a student.

```
import java.util.Scanner;

class Subject

{

        int subj_marks;

        int grade;

        int credits;

}

class Student
```

```
{
        String name;
        String usn;
        double SGPA;
        Scanner s;
        Subject subj[];
        Student()
        {
                int i;
                subj=new Subject[8];
                for(i=0;i<8;i++)
                        subj[i]=new Subject();
                s=new Scanner(System.in);
        }
        void getStudentDetails()
        {
                System.out.print("Enter Student Name: ");
                name=s.next();
                System.out.print("Enter USN: ");
                usn=s.next();
        }
        void getMarks()
        {   s=new Scanner(System.in);
                for(int i=0;i<8;i++)
                {
```

```
System.out.println("Enter details for subject "+(i+1)+":");
System.out.print("Enter marks of student: ");
subj[i].subj_marks=s.nextInt();
System.out.print("Enter credits of subject: ");
subj[i].credits=s.nextInt();
if(subj[i].subj_marks>=90)
{
        subj[i].grade=10;
}
else if(subj[i].subj_marks>=80)
{
        subj[i].grade=9;
}
else if(subj[i].subj_marks>=70)
{
        subj[i].grade=8;
}
else if(subj[i].subj_marks>=60)
{
        subj[i].grade=7;
}
else if(subj[i].subj_marks>=50)
{
        subj[i].grade=6;
}
```

```
                else if(subj[i].subj_marks>=40)
                {
                        subj[i].grade=5;
                }
                else
                {
                        subj[i].grade=0;
                }
        }
}
void computeSGPA()
{
        double totalcredits=0;
        double weightedSum=0;
        for(int i=0;i<8;i++)
        {
                totalcredits+=subj[i].credits;
                weightedSum+=subj[i].grade*subj[i].credits;
        }
        SGPA=weightedSum/totalcredits;
}
void DisplayDetails()
{
        System.out.println("\nStudent Details:");
        System.out.println("Name: "+name);
```

```
                System.out.println("USN: "+usn);
                System.out.println("SGPA: "+SGPA);
        }
}
public class StudentSGPAMain
{
        public static void main(String args[])
        {
                Student s1= new Student();
                s1.getStudentDetails();
                s1.getMarks();
                s1.computeSGPA();
                s1.DisplayDetails();
        }
}
```

Output:

```
C:\Users\preet\OneDrive\Desktop\3rd Sem Java Pgms>java StudentSGPAMain
Enter Student Name: Preethi
Enter USN: 1BM22CS209
Enter details for subject 1:
Enter marks of student: 56
Enter credits of subject: 3
Enter details for subject 2:
Enter marks of student: 89
Enter credits of subject: 4
Enter details for subject 3:
Enter marks of student: 100
Enter credits of subject: 4
Enter details for subject 4:
Enter marks of student: 90
Enter credits of subject: 3
Enter details for subject 5:
Enter marks of student: 88
Enter credits of subject: 3
Enter details for subject 6:
Enter marks of student: 99
Enter credits of subject: 2
Enter details for subject 7:
Enter marks of student: 77
Enter credits of subject: 2
Enter details for subject 8:
Enter marks of student: 99
Enter credits of subject: 1

Student Details:
Name: Preethi
USN: 1BM22CS209
SGPA: 8.954545454545455
```

Lab 3

Create a class Book which contains four members: name, author, price, num_pages. Include a constructor to set the values for the members. Include methods to set and get the details of the objects. Include a toString( ) method that could display the complete details of the book. Develop a Java program to create n book objects.

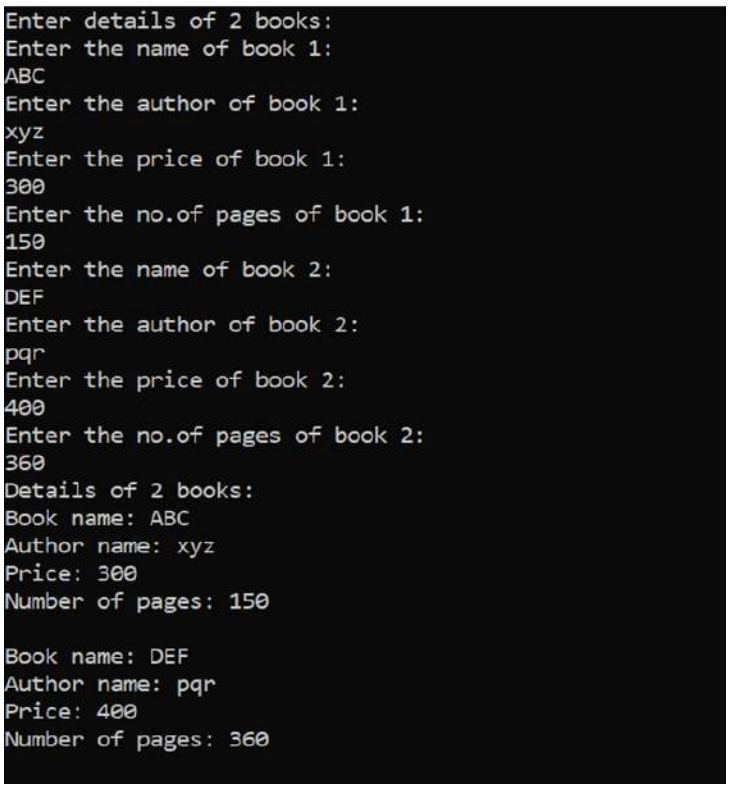
import java.util.Scanner;

```
class Books
{
	String name;
	String author;
	int price;
	int num_pages;
	Books(String name,String author,int price,int num_pages)
	{
		this.name=name;
		this.author=author;
		this.price=price;
		this.num_pages=num_pages;
	}
	public String toString()
	{
		String name,author,price,num_pages;
		name = "Book name: " + this.name + "\n";
		author = "Author name: " + this.author + "\n";
		price = "Price: " + this.price + "\n";
		num_pages = "Number of pages: " + this.num_pages + "\n";
		return name+author+price+num_pages;
	}
}
class BooksMain
{
```

```
public static void main(String args[])
{
        Scanner s=new Scanner(System.in);
        int n;
        String name;
        String author;
        int price;
        int num_pages;
        System.out.println("Enter the no.of books: ");
        n=s.nextInt();
        Books[] b= new Books[n];
//Books b[]; //array of objects
        //b=new Books[n];
        System.out.println("Enter details of "+n+" books:");
        for(int i=0;i<n;i++)
        {
                System.out.println("Enter the name of book "+(i+1)+":");
                name=s.next();
                System.out.println("Enter the author of book "+(i+1)+":");
                author=s.next();
                System.out.println("Enter the price of book "+(i+1)+":");
                price=s.nextInt();
                System.out.println("Enter the no.of pages of book "+(i+1)+":");
                num_pages=s.nextInt();
                b[i]=new Books(name,author,price,num_pages);
```

```
            }
            System.out.println("Details of "+n+" books:");
            for(int i=0;i<n;i++)
            {
                System.out.println(b[i].toString());
            }
        }
}
```

Output:



Lab 4

Develop a Java program to create an abstract class named Shape that contains two integers and an empty method named printArea( ). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the classShape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.

```
import java.util.Scanner;
abstract class InputScanner
{
	abstract void get_dim();
}
abstract class Shape extends InputScanner
{
	int a,b;
	Shape(int a,int b)
	{
		this.a=a;
		this.b=b;
	}
	Shape(int a)
	{
		this.a=a;
		this.b=0;
	};
	abstract void printArea();
```

```
}
class Rectangle extends Shape
{
	Rectangle(int a,int b)
	{
		super(a,b);
	}
	void get_dim()
	{
		System.out.println("Enter the dimensions of the rectangle(length and breadth)");
		Scanner s=new Scanner(System.in);
		a=s.nextInt();
		b=s.nextInt();
	}
	void printArea()
	{
		System.out.println("Area of Rectangle = "+(a*b));
	}
}
class Triangle extends Shape
{
	Triangle(int a,int b)
	{
		super(a,b);
	}
```

```
	void get_dim()
	{
		System.out.println("Enter the dimensions of the triangle(base and height)");
		Scanner s=new Scanner(System.in);
		a=s.nextInt();
		b=s.nextInt();
	}
	void printArea()
	{
		System.out.println("Area of triangle = "+(0.5*a*b));
	}
}
class Circle extends Shape
{
	Circle(int a)
	{
		super(a);
	}
	void get_dim()
	{
		System.out.println("Enter the dimension of the circle(radius)");
		Scanner s=new Scanner(System.in);
		a=s.nextInt();

	}
```

```
	void printArea()
	{
		System.out.println("Area of circle = "+(3.14*a*a));
	}
}
class ShapeMain
{
	public static void main(String args[])
	{
		Rectangle r= new Rectangle(0,0);
		Triangle t= new Triangle(0,0);
		Circle c= new Circle(0);
		r.get_dim();
		t.get_dim();
		c.get_dim();
		r.printArea();
		t.printArea();
		c.printArea();
	}
}
```

Output:

```
C:\Users\preet\OneDrive\Desktop\3rd Sem Java Pgms>java ShapeMain
Enter the dimensions of the rectangle(length and breadth)
10 2
Enter the dimensions of the triangle(base and height)
2 3
Enter the dimension of the circle(radius)
3
Area of Rectangle = 20
Area of triangle = 3.0
Area of circle = 28.259999999999998
```

Lab 5

Develop a Java program to create a class Bank that maintains two kinds of account for its customers, one called savings account and the other current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed. Create a class Account that stores customer name, account number and type of account. From this derive the classes Cur-acct and Sav-acct to make them more specific to their requirements. Include the necessary methods in order to achieve the following tasks:

a) Accept deposit from customer and update the balance.

b) Display the balance.

c) Compute and deposit interest

d) Permit withdrawal and update the balance

Check for the minimum balance, impose penalty if necessary and update the balance.

```
import java.util.Scanner;
class Account {
    String cust_name;
    int Accno;
    double balance = 0;
    public void get_details() {
            Scanner scanner = new Scanner(System.in);
            System.out.println("Enter the customer name: ");
            cust_name = scanner.next();
            System.out.println("Enter the account number: ");
            Accno = scanner.nextInt();
    }
    public void display_details(String Acc_type) {
            System.out.println("Customer Name: " + cust_name);
            System.out.println("Account Number: " + Accno);
            System.out.println("Account Type: " + Acc_type);
            System.out.println("Balance: " + balance);
    }
}
class Sav_acc extends Account {
    double interestRate = 0.05; // Compound interest rate for savings account
    public Sav_acc(double balance) {
            this.balance = balance;
    }
    public void deposit_s() {
```

```
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the deposit amount: ");
        double amount = s.nextDouble();
        balance += amount;
        System.out.println("Deposit of " + amount + " successful.");
    }
    public void computeInterest() {
        double interest = balance * interestRate;
        balance += interest;
        System.out.println("Interest of " + interest + " computed and added to the account.");
    }
    public void withdraw_s() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the withdrawal amount: ");
        double amt = s.nextDouble();
        if (balance >= amt) {
        balance -= amt;
        System.out.println("Withdrawal of " + amt + " successful.");
        }
        else {
        System.out.println("Insufficient funds. Withdrawal not permitted.");
        }
    }
}
class Cur_acc extends Account {
```

```
public Cur_acc(double balance) {
        this.balance = balance;
}
public void deposit_c() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the deposit amount: ");
        double amount = s.nextDouble();
        balance += amount;
        System.out.println("Deposit of " + amount + " successful.");
}
public void Cheque_book() {
        String phone_no;
        String address;
        Scanner s = new Scanner(System.in);
        System.out.println("Enter your phone number: ");
        phone_no = s.next();
        System.out.println("Enter your address: ");
        address = s.next();
        System.out.println("Cheque book request processed successfully.");
}
public void check_min_bal() {
        double min_bal = 1000;
        double service_charge = 50;
        if (balance < min_bal) {
        balance -= service_charge;
```

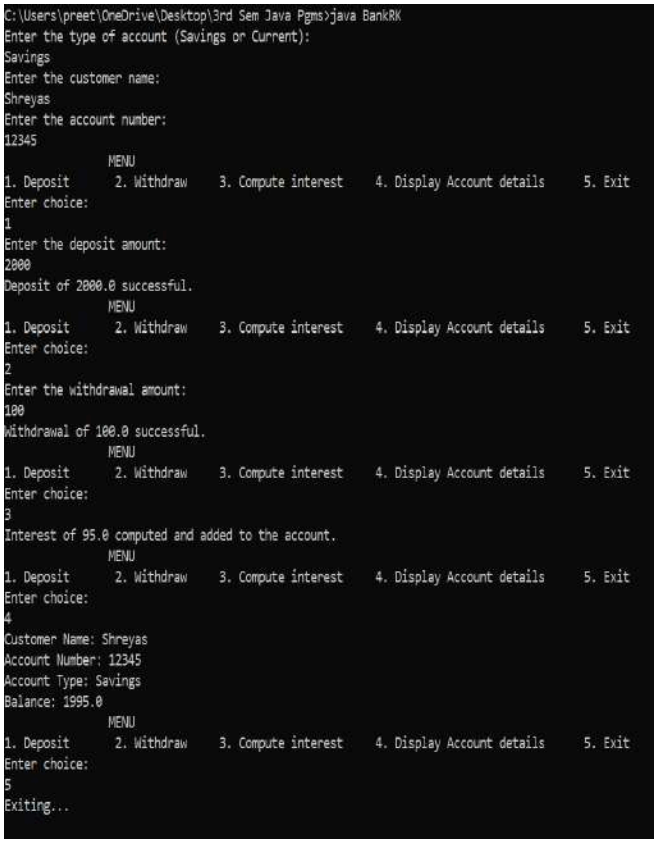
```
        System.out.println("Service charge of $" + service_charge + " imposed due to balance falling below minimum.");
        } else {
        System.out.println("Current balance is valid");
        }
    }

    public void withdraw_c() {
        Scanner s = new Scanner(System.in);
        System.out.println("Enter the withdrawal amount: ");
        double amt = s.nextDouble();
        if (balance >= amt) {
        balance -= amt;
        System.out.println("Withdrawal of " + amt + " successful.");
        } else {
        System.out.println("Insufficient funds. Withdrawal not permitted.");
        }
    }
}

class BankRK {
    public static void main(String[] args) {
        Scanner p = new Scanner(System.in);
        System.out.println("Enter the type of account (Savings or Current): ");
        String Acc_type = p.next();
```

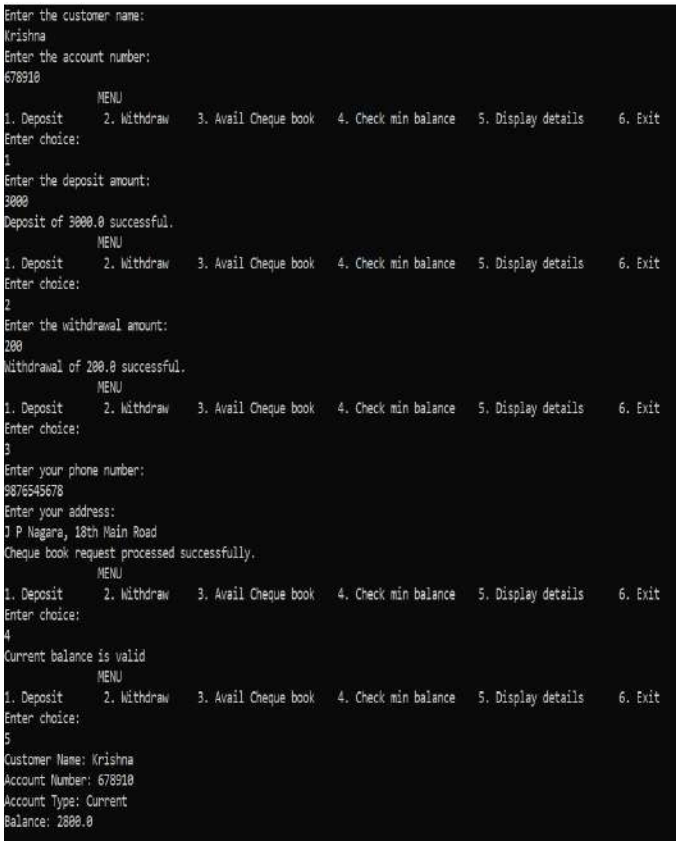
```
if (Acc_type.equals("Savings")) {
Sav_acc sa = new Sav_acc(0); // Initialize with zero balance
sa.get_details();
int choice;
do {
   System.out.println("\t\tMENU\t\t");
        System.out.println("1. Deposit\t 2. Withdraw\t 3. Compute interest\t 4. Display
Account details\t 5. Exit");
        System.out.println("Enter choice: ");
        choice = p.nextInt();
        switch (choice) {
        case 1:
        sa.deposit_s();
        break;
        case 2:
        sa.withdraw_s();
        break;
        case 3:
        sa.computeInterest();
        break;
        case 4:
        sa.display_details(Acc_type);
        break;
        case 5:
        System.out.println("Exiting...");
        break;
```

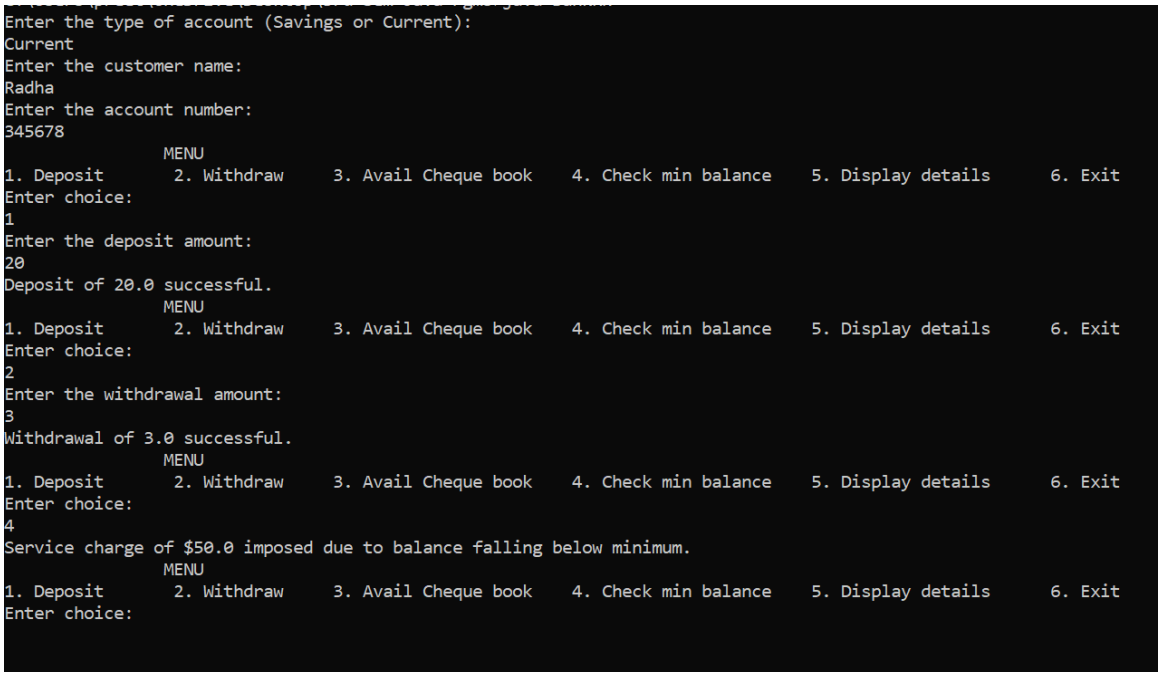
```
                default:
                System.out.println("Invalid choice");
                }
        } while (choice != 5);
        } else if (Acc_type.equals("Current")) {
        Cur_acc ca = new Cur_acc(0); // Initialize with zero balance
        ca.get_details();

        int choice;
        do {
           System.out.println("\t\tMENU\t\t");
                System.out.println("1. Deposit\t 2. Withdraw\t 3. Avail Cheque book \t 4. Check
min balance\t 5. Display details\t 6. Exit");
                System.out.println("Enter choice: ");
                choice = p.nextInt();
                switch (choice) {
                case 1:
                ca.deposit_c();
                break;
                case 2:
                ca.withdraw_c();
                break;
                case 3:
                ca.Cheque_book();
                break;
                case 4:
```

```
                ca.check_min_bal();
                break;
                case 5:
                ca.display_details(Acc_type);
                break;
                case 6:
                System.out.println("Exiting...");
                break;
                default:
                System.out.println("Invalid choice");
                }
        } while (choice != 6);
        } else {
        System.out.println("Invalid account type.");
        }
    }
}
```

Output:


C:\Users\preet\OneDrive\Desktop\3rd Sem Java Pgms>java BankRK
Enter the type of account (Savings or Current):
Savings
Enter the customer name:
Shreyas
Enter the account number:
12345
MENU
1. Deposit 2. Withdraw 3. Compute interest 4. Display Account details 5. Exit
Enter choice:
1
Enter the deposit amount:
2000
Deposit of 2000.0 successful.
MENU
1. Deposit 2. Withdraw 3. Compute interest 4. Display Account details 5. Exit
Enter choice:
2
Enter the withdrawal amount:
100
Withdrawal of 100.0 successful.
MENU
1. Deposit 2. Withdraw 3. Compute interest 4. Display Account details 5. Exit
Enter choice:
3
Interest of 95.0 computed and added to the account.
MENU
1. Deposit 2. Withdraw 3. Compute interest 4. Display Account details 5. Exit
Enter choice:
4
Customer Name: Shreyas
Account Number: 12345
Account Type: Savings
Balance: 1995.0
MENU
1. Deposit 2. Withdraw 3. Compute interest 4. Display Account details 5. Exit
Enter choice:
5
Exiting...


Enter the customer name:
Krishna
Enter the account number:
678910
MENU
1. Deposit 2. Withdraw 3. Avail Cheque book 4. Check min balance 5. Display details 6. Exit
Enter choice:
1
Enter the deposit amount:
3000
Deposit of 3000.0 successful.
MENU
1. Deposit 2. Withdraw 3. Avail Cheque book 4. Check min balance 5. Display details 6. Exit
Enter choice:
2
Enter the withdrawal amount:
200
Withdrawal of 200.0 successful.
MENU
1. Deposit 2. Withdraw 3. Avail Cheque book 4. Check min balance 5. Display details 6. Exit
Enter choice:
3
Enter your phone number:
9876545678
Enter your address:
J P Nagara, 18th Main Road
Cheque book request processed successfully.
MENU
1. Deposit 2. Withdraw 3. Avail Cheque book 4. Check min balance 5. Display details 6. Exit
Enter choice:
4
Current balance is valid
MENU
1. Deposit 2. Withdraw 3. Avail Cheque book 4. Check min balance 5. Display details 6. Exit
Enter choice:
5
Customer Name: Krishna
Account Number: 678910
Account Type: Current
Balance: 2800.0


Enter the type of account (Savings or Current):
Current
Enter the customer name:
Radha
Enter the account number:
345678
MENU
1. Deposit 2. Withdraw 3. Avail Cheque book 4. Check min balance 5. Display details 6. Exit
Enter choice:
1
Enter the deposit amount:
20
Deposit of 20.0 successful.
MENU
1. Deposit 2. Withdraw 3. Avail Cheque book 4. Check min balance 5. Display details 6. Exit
Enter choice:
2
Enter the withdrawal amount:
3
Withdrawal of 3.0 successful.
MENU
1. Deposit 2. Withdraw 3. Avail Cheque book 4. Check min balance 5. Display details 6. Exit
Enter choice:
4
Service charge of $50.0 imposed due to balance falling below minimum.
MENU
1. Deposit 2. Withdraw 3. Avail Cheque book 4. Check min balance 5. Display details 6. Exit
Enter choice:

Lab 6

Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE which has the class External which is a derived class of Student. This class has an array that stores the SEE marks scored in five courses of the current semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

```
// Package CIE

package CIE;

import java.util.Scanner;

public class student

{

	protected String usn = new String();

	protected String name = new String();

	protected int sem;

	public void inputDetails()

	{

			Scanner s=new Scanner(System.in);

			System.out.println("Enter student usn:");

			usn = s.next();

			System.out.println("Enter student name:");

			name = s.next();

			System.out.println("Enter student sem:");

			sem = s.nextInt();
```

```
        }
        public void displayDetails()
        {
                        System.out.println("Student Details:\n");
                        System.out.println("USN: "+usn);
                        System.out.println("Name: "+name);
                        System.out.println("Sem: "+sem);
        }
}
public class internals extends student
{
        protected int marks[] = new int[5];
        public void CIEmarks()
        {
                        Scanner s= new Scanner(System.in);
                        for(int i=0;i<5;i++)
                        {
                                System.out.println("Subject "+(i+1)+" marks");
                                marks[i] = s.nextInt();
                        }
        }
}

// Package SEE
package SEE;
```

```
import CIE.internals;
import java.util.Scanner;
public class externals extends internals
{
	protected int marks[];
	protected int finalMarks[];
	public externals()
	{
			marks = new int[5];
			finalMarks = new int[5];
	}
	public void SEEMarks()
	{
			Scanner s= new Scanner(System.in);
			for(int i=0;i<5;i++)
			{
				System.out.println("Subject "+(i+1)+" marks:");
				marks[i] = s.nextInt();
			}
	}
	public void calMarks()
	{
			for(int i=0;i<5;i++)
				finalMarks[i] = marks[i]/2 + super.marks[i];
	}
```

```
	public void displayFinalMarks()
	{
			for(int i=0;i<5;i++)
				System.out.println("Subject "+(i+1)+": "+finalMarks[i]);
	}
}

// Class Main
import SEE.externals;
class Main
{
	public static void main(String args[])
	{
			int num = 2;
			externals finalMarks[] = new externals[num];
			for(int i=0;i<num;i++)
			{
				finalMarks[i] = new externals();
				finalMarks[i].inputDetails();
				System.out.println("Enter CIE marks:");
				finalMarks[i].CIEmarks();
				System.out.println("Enter SEE marks:");
				finalMarks[i].SEEMarks();
			}
```

```
                System.out.println("Displaying data:\n");
                for (int i=0;i<num;i++)
                {
                        finalMarks[i].calMarks();
                        finalMarks[i].displayDetails();
                        finalMarks[i].displayFinalMarks();
                }
        }
}
```

Output:

```
C:\Users\preet\OneDrive\Desktop\3rd Sem Java Pgms>java Main
Enter USN: 1BM22CS207
Enter Name: Preethi
Enter Semester: 3
Enter CIE marks
Enter CIE marks for :
Enter marks for Subject 1: 45
Enter marks for Subject 2: 50
Enter marks for Subject 3: 37
Enter marks for Subject 4: 49
Enter marks for Subject 5: 44
Enter SEE marks
Subject 1 marks: 90
Subject 2 marks: 99
Subject 3 marks: 97
Subject 4 marks: 90
Subject 5 marks: 87
```

```
USN: 1BM22CS207
Name: Preethi
Semester: 3
Subject 1: 90
Subject 2: 99
Subject 3: 85
Subject 4: 94
Subject 5: 87
```

Lab 7

Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends the base class. In Father class, implement a constructor which takes the age and throws the exception WrongAge( ) when the input age<0. In Son 25 class, implement a constructor that cases both father and son's age and throws an exception if son's age is >=father's age.

```
import java.util.*;
class wrongAge extends Exception
{
	wrongAge(String s)
	{
		super(s);
	}
}
class input
{
	int f_age;
```

```
        int s_age;
        Scanner sc= new Scanner(System.in);
}
class father extends input
{
        father() throws wrongAge
        {
                System.out.println("Enter father's age:");
                f_age = sc.nextInt();
                if(f_age < 0)
                        throw new wrongAge("Age cannot be negative");
        }
        void display1()
        {
                System.out.println("Father's age is : "+f_age);
        }
}
class son extends father
{
        son() throws wrongAge
        {
                System.out.println("Enter son's age:");
                s_age = sc.nextInt();
                if (s_age > f_age)
                {
```

```
                throw new wrongAge("Son's age cannot be greater than father's age");
            }
            else if (s_age <0)
                throw new wrongAge("Age cannot be negative");
        }
        void display2()
        {
            System.out.println("Son's age is : "+s_age);
        }
}
class MainSon
{
public static void main (String args[]){
        try
        {
            son s =new son();
            s.display1();
            s.display2();
        }
        catch(wrongAge e)
        {
            System.out.println("Error : "+e);
        }
}
```

Output:

```
C:\Users\preet\OneDrive\Desktop\3rd Sem Java Pgms>java MainSon
Enter father's age:
34
Enter son's age:
45
Error : wrongAge: Son's age cannot be greater than father's age

C:\Users\preet\OneDrive\Desktop\3rd Sem Java Pgms>javac MainSon.java

C:\Users\preet\OneDrive\Desktop\3rd Sem Java Pgms>java MainSon
Enter father's age:
-9
Error : wrongAge: Age cannot be negative

C:\Users\preet\OneDrive\Desktop\3rd Sem Java Pgms>javac MainSon.java

C:\Users\preet\OneDrive\Desktop\3rd Sem Java Pgms>java MainSon
Enter father's age:
45
Enter son's age:
-8
Error : wrongAge: Age cannot be negative

C:\Users\preet\OneDrive\Desktop\3rd Sem Java Pgms>javac MainSon.java

C:\Users\preet\OneDrive\Desktop\3rd Sem Java Pgms>java MainSon
Enter father's age:
50
Enter son's age:
20
Father's age is : 50
Son's age is : 20
```

Lab 8

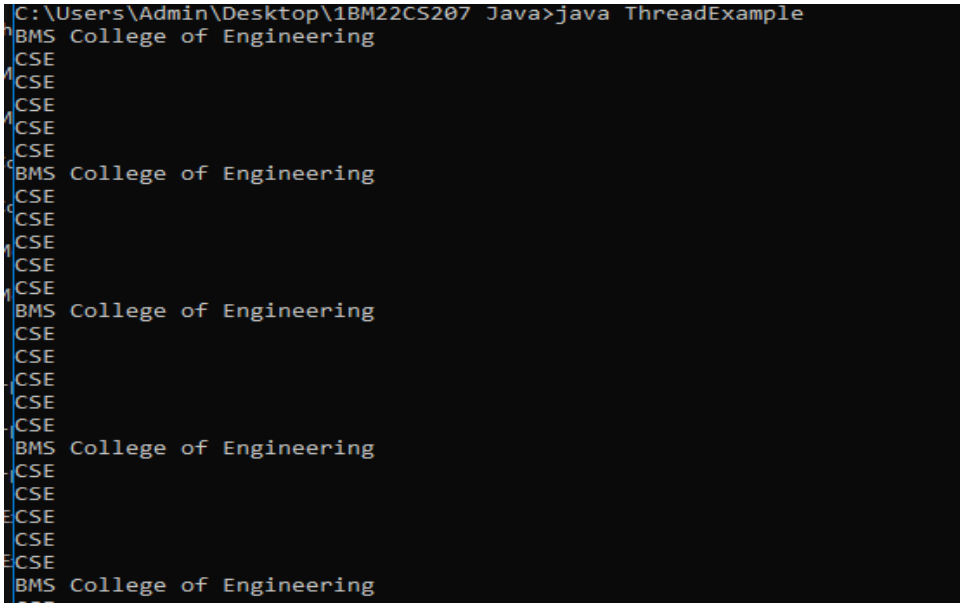
Write a program which creates two threads, one thread displaying “BMS College of Engineering” once every ten seconds and another displaying “CSE” once every two seconds. class DisplayThread extends Thread.

```
import java.util.Scanner;

class DisplayThread extends Thread {

    private String message;

    private int interval;
```

```
    public DisplayThread(String message, int interval) {
        this.message = message;
        this.interval = interval;
        // No need to initialize isRunning here
    }
    @Override
    public void run() {
        while (true) {
            System.out.println(message);
            try {
                Thread.sleep(interval * 1000); // Convert seconds to milliseconds
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    }
}
public class ThreadExample {
    public static void main(String[] args) {
        // Create threads with different messages and intervals
        DisplayThread thread1 = new DisplayThread("BMS College of Engineering", 10);
        DisplayThread thread2 = new DisplayThread("CSE", 2);
        // Start the threads
        thread1.start();
        thread2.start();
```

```
    }
}
```

Output:



Lab 9

Write a program that creates a user interface to perform integer divisions. The user enters two numbers in the text fields, Num1 and Num2. The division of Num1 and Num2 is displayed in the Result field when the Divide button is clicked. If Num1 or Num2 were not an integer, the program would throw a NumberFormatException. If Num2 were Zero, the program would throw an Arithmetic Exception Display the exception in a message dialog box.

```
import javax.swing.*;

import java.awt.*;
```

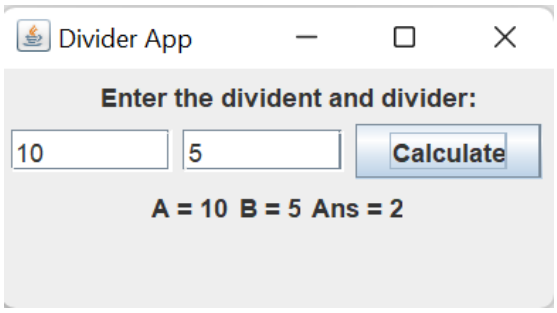
```
import java.awt.event.*;
class SwingDemo{
    SwingDemo(){
        // create jframe container
        JFrame jfrm = new JFrame("Divider App");
        jfrm.setSize(275, 150);
        jfrm.setLayout(new FlowLayout());
        // to terminate on close
        jfrm.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // text label
        JLabel jlab = new JLabel("Enter the divident and divider:");
        // add text field for both numbers
        JTextField ajtf = new JTextField(8);
        JTextField bjtf = new JTextField(8);
        // calc button
        JButton button = new JButton("Calculate");
        // labels
        JLabel err = new JLabel();
        JLabel alab = new JLabel();
        JLabel blab = new JLabel();
        JLabel anslab = new JLabel();
        // add in order :)
        jfrm.add(err);  // to display error bois
```

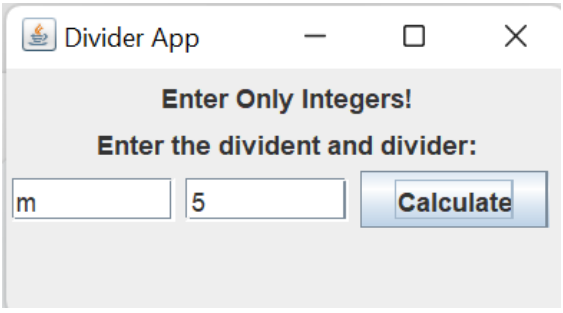
```
jfrm.add(jlab);
jfrm.add(ajtf);
jfrm.add(bjtf);
jfrm.add(button);
jfrm.add(alab);
jfrm.add(blab);
jfrm.add(anslab);
ActionListener l = new ActionListener() {
   public void actionPerformed(ActionEvent evt) {
      System.out.println("Action event from a text field");
   }
};
ajtf.addActionListener(l);
bjtf.addActionListener(l);
button.addActionListener(new ActionListener() {
   public void actionPerformed(ActionEvent evt) {
      try{
         int a = Integer.parseInt(ajtf.getText());
         int b = Integer.parseInt(bjtf.getText());
         int ans = a/b;
         alab.setText("\nA = " + a);
         blab.setText("\nB = " + b);
         anslab.setText("\nAns = "+ ans);
```

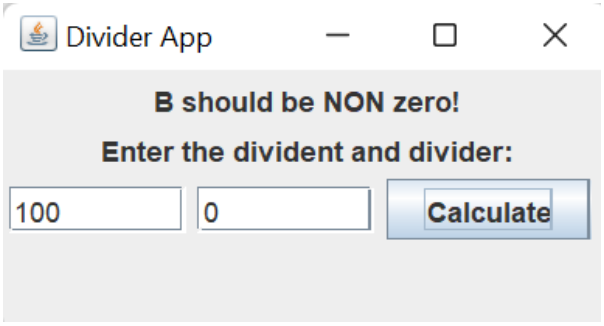
```
            }
            catch(NumberFormatException e){
                alab.setText("");
                blab.setText("");
                anslab.setText("");
                err.setText("Enter Only Integers!");
            }
            catch(ArithmeticException e)
           {
                alab.setText("");
                blab.setText("");
                anslab.setText("");
                err.setText("B should be NON zero!");
            }
        }
    });

    // display frame
    jfrm.setVisible(true);
}
public static void main(String args[]){
    // create frame on event dispatching thread
    SwingUtilities.invokeLater(new Runnable(){
```

```
            public void run(){
                new SwingDemo();
            }
        });
    }
}
```

Output:







Lab 10

Demonstrate Inter process Communication and deadlock.

InterProcess Communication:

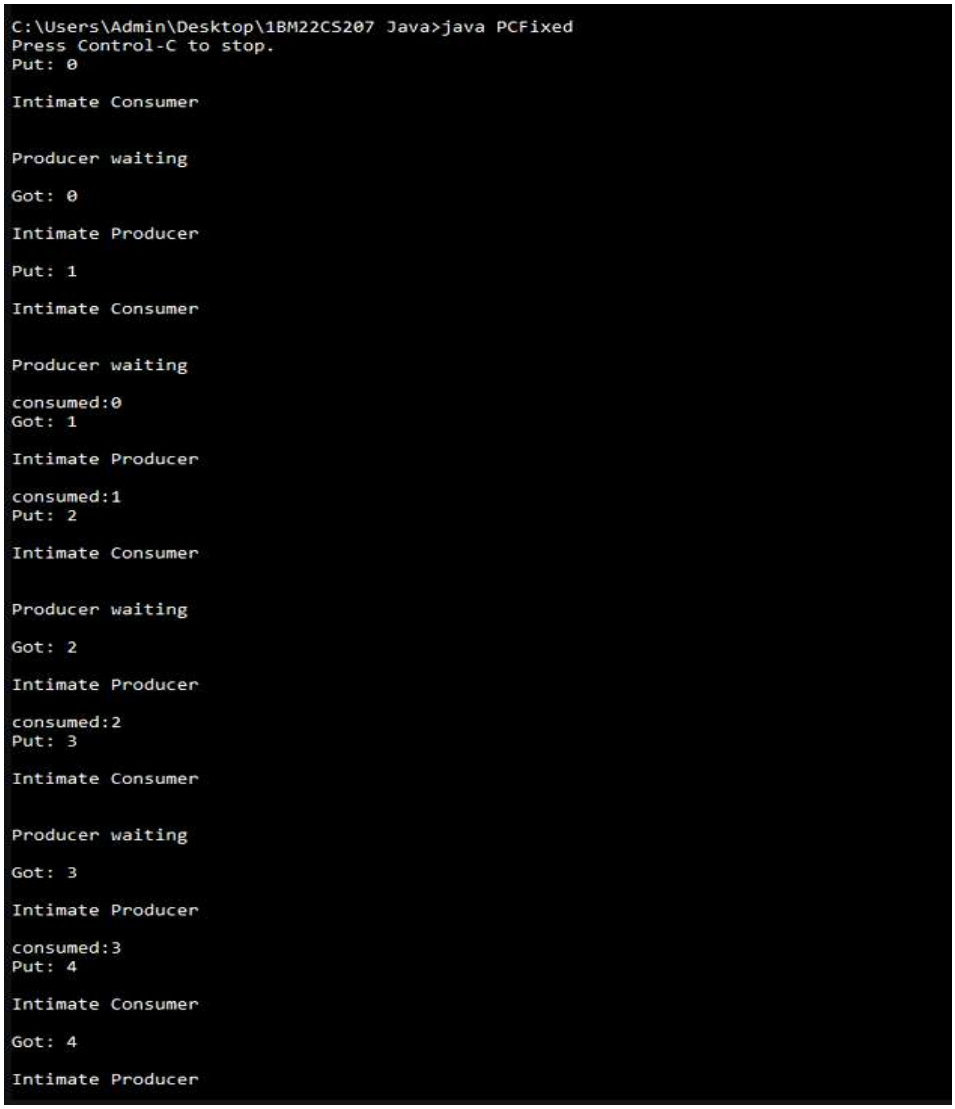
```
class Q
{
        int n;
        boolean valueSet = false;
        synchronized int get()
        {
                        while(!valueSet)
                        try {
                        System.out.println("Consumer waiting");
                        wait();
        } catch(InterruptedException e) {
        System.out.println("InterruptedException caught");
        }
        System.out.println("Got: " + n);
        valueSet = false;
        System.out.println("Intimate Producer");
        notify();
        return n;
        }

        synchronized void put(int n) {
        while(valueSet)
```

```
        try {
        System.out.println("Producer waiting");
        wait();
        } catch(InterruptedException e) {
        System.out.println("InterruptedException caught");
        }
        this.n = n;
        valueSet= true;
        System.out.println("Put: " + n);
        System.out.println("Intimate Consumer");
        notify();
        }
}
class Producer implements Runnable {
        Q q;
        Producer(Q q) {
        this.q = q;
        new Thread(this, "Producer").start();
        }
        public void run() {
        int i = 0;
        while(i<5) {
        q.put(i++);
```

```
        }
        }
}
class Consumer implements Runnable {
        Q q;
        Consumer(Q q) {
        this.q = q;
        new Thread(this, "Consumer").start();
        }
        public void run() {
        int i=0;
        while(i<5) {
        int r=q.get();
        i++;
        }
        }
}
class PCFixed {
        public static void main(String args[]) {
        Q q = new Q();
        new Producer(q);
        new Consumer(q);
        System.out.println("Press Control-C to stop.");
```

```
        }
}
```

Output:



Deadlock:

```
class A {
synchronized void foo(B b) {
```

```
String name =
Thread.currentThread().getName();
System.out.println(name + " entered A.foo");
try {
Thread.sleep(1000);
} catch(Exception e) {
System.out.println("A Interrupted");
}
System.out.println(name + " trying to call B.last()");
b.last();
}
void last() {
System.out.println("Inside A.last");
}
}
class B {
synchronized void bar(A a) {
String name = Thread.currentThread().getName();
System.out.println(name+"entered B.bar");
try {
Thread.sleep(1000);
} catch(Exception e) {
System.out.println("B Interrupted");
}
System.out.println(name + " trying to call A.last()");
```

```
a.last();
}
void last() {
System.out.println("Inside A.last");
}
}
class Deadlock implements Runnable
{
A a = new A();
B b = new B();
Deadlock() {
Thread.currentThread().setName("MainThread");
Thread t = new Thread(this,"RacingThread");
t.start();
a.foo(b);
System.out.println("Back in main thread");
}
public void run() {
b.bar(a);
System.out.println("Back in other thread");
}
public static void main(String args[]) {
new Deadlock();
}
}
```

Output:

```
C:\Users\preet\OneDrive\Desktop\3rd Sem Java Pgms>java Deadlock
RacingThreadentered B.bar
MainThread entered A.foo
RacingThread trying to call A.last()
MainThread trying to call B.last()
Inside A.last
Inside A.last
Back in main thread
Back in other thread

C:\Users\preet\OneDrive\Desktop\3rd Sem Java Pgms>
```